Wire TAPS

THE NEWSLETTER OF THE ATARI PROGRAMMERS' SOCIETY

SUMMER & SEPTEMBER 1986 LAST ISSUE

All Good Things Must Come to an End ...or Must They?

by Nick Coleman

The first part of this article is about the "End".

It had to happen sooner or later. Your long-standing executive has finally decided to call it a day, and with no heirs apparent, have also decided to close the TAPS umbrella. We hope you have enjoyed the TAPS experience and have benefited from WireTAPS, the Software Library and our three monthly meetings.

Some thanks are in order to the various members of the executive, past and present. Though individual words of appreciation for all who have contributed over the years, is appropriate, there is just not enough space to do so. At the risk of slighting some however, a special note of thanks must be given to those who were responsible for TAPS two major institutions, the Newsletter and the Software Library.

Peter Blake, the editor of WireTAPS has perhaps donated more of his time to TAPS than any other. He has been solely responsible for the newsletter. Most and sometimes all the articles are his creation. Those that are not are at the very least edited, typed in or converted to Writer's Tool, with which this newsletter is written. The work does not stop there however. The final draft is taken to the copy shop for printing, the 200 or so copies carried home, stamps and mailing labels applied, new membership cards or renewal notices and reminders as appropriate are included, the whole lot folded and taped together and only then taken to the post office, hopefully in sufficient time to reach each of you before the first meeting of the month. This is a labourious task for which Peter should be commended, not just for the excellence in his work but for his faithfulness to the club, when no relief was in sight.

Tim Glass, Chairman of our Public Domain Library, together with Doug Barton, Tony Rainone, Chris Thomsen, Bruce Fish and their latest help Dave Chasson, have all worked hard to assemble interesting software, fill backorders and without fail, create a new disk each month. As a previous member and chairman of this committee, I can assure you this is no small task and there have been many long nights spent copying disks for the club. Without the Library's contribution, TAPS would have faltered long ago.

Others that have played significant roles in the evolution and maintenance of TAPS (these are in alphabetical order and I appologize for any omissions, incorrect spelling and not writing a paragraph for each) are:
Ed Apostel, Larry Breakwell, Eric Cadesky, Dave Castell, Jordan Chilcott, the Cruse Family, Ian Davidson, Terry Demerson (alias Kevin Fica), Joe Fodor, Ralph Frampton, Jeff Greenway, Ray Hudson, Roger Humphries, Vern Jackson, Sean Keeley, George Kettela, Jim Kuprowsky, Ken Lloyd, Ron Martin, Alan Monroe, John Reeves, Hans Reymer, Bob & Kim Rowcliffe, Paul Scudds, Fabian Tapp, and last but by no means least, two of our very most important contributors, Graham Thorley and Brenda Tupling.

We also received a little professional help (the club did, not the executive, though we've often felt we were a little crazy) and wish to thank Ron Adams, Ian Chadwick, Marty Herzog, David Langdon, Barry Pusitz, Greg Weir, the Red Baron BBS, the Sports BBS, Toronto Computes and the fellows at Electronics Playworld that started it all.

If this had been a meeting rather than an article, a round of applause and standing ovation would now be in order; ...thank you.

I hope by now the initial shock about the "End" has worn off a bit, because at this point I would like to provide a short dissertation on the "... or Must They?", part of the title. You are all probably wondering what the cheque attached to the newsletter is for. Well it's for money, and unlike some fly-by-night fitness clubs, it amounts to an almost complete refund for each month left before your TAPS membership was due to expire. If your membership expires this month anyway however, you obviously didn't get a cheque.

Now that you have this new found wealth, what are you going to do with it?

(At this point, had you been listening to background music, it would be appropriate if some sort of drum roll was to occur, as if in great expectation, something of vital importance was about to be announced.)

Rather than leaving the membership alone in the cold, we have arranged to keep the meeting locations available until Christmas (an early present), but if no further arrangements have been made by that time, then you will indeed be left alone in the cold. Unlike September, it snows in December ya know!

Now let's recap. (1) TAPS as an umbrella organization, is gone. (2) It therefore follows that none of you belong to an Atari Users' Group [some are also members of TAF, Ed.]. (3) Most of you either have a refund or were about to renew your membership. (4) There are 3 monthly meeting locations available until Christmas.

At the risk of being horribly obvious, it might be suggested that you continue to meet just as before, as if nothing has really changed, which actually, it hasn't. Well maybe a little.

Despite the lack of a co-ordinating executive, a common Newsletter and the Software Library (though each group will at least have an up-to-date set of TAPS Library Disks), it is quite apparent there is a very good opportunity and every likelihood the clubs will continue, but with new names and perhaps new directions. For instance, how about the Atari Legion of Scarborough Ontario (ALSO), the Society of Oshawa Atari Programmers (SOAP), and the Etobicoke Atari Recreational Society (EARS). This would make a nice clean wash of things and ALSO SOAP EARS.

Though unlikely, I realize there just may be other possibilities that might provide slightly more acceptable acronyms.

For those living in the Willowdale area, you may also wish to consider joining TAF (the Toronto Atari Federation) as well or as an alternative. There should be an invitation to all TAPS members from TAF, somewhere else in this issue.

I want to make it clear however, that despite the unanimous decision at our August executive meeting to wind down TAPS, each group also indicated quite strongly that they wished to continue in some form or other and you are encouraged to do so. There may be (who knows?) less of an attempt to run a structured meeting, with a move towards the successful and informal arrangement found at Scarborough, where Atari owners assemble to chat about Atari news, problems and ideosyncracies.

Toronto's (Etobicoke) September meeting, as usual for a September meeting, will not be held at Norseman. The alternative this time will be Humbercrest Public School near Jane and Bloor. Those attending will decide whether they prefer Norseman or Humbercrest for their regular meeting location.

Scarborough's September meeting will be held (just this once) on the second **Wednesday** instead of Tuesday, due to a conflict at the Cedar Ridge Creative Centre.

At the time of printing, Oshawa's meeting will still be the second Saturday of the month, at the Whitby Public Library.

Well that's about it, I think. I've enjoyed (most of the time) being part of TAPS over the many years since its inception, and have every confidence the 4 groups that were established will continue to flourish (Peterborough, the 4th group, is already independent and doing well). It's all up to you now, though no doubt, you'll still find us at most of the meetings keeping a motherly eye on things...so long and good luck...

EDITORIAL

By Peter BLAKE

Well, I guess that I should tell you that I finally broke down and bought an ST. A TAPS' member made me an offer that I just could not refuse. So I am now the owner of a 1040ST. As I spend most of my time programming and word processing, I opted for the monochrome monitor. In the months to come, I shall be learning more about

the ST, but for the moment I have not had the time to do very much with it.

It is like the first days that I had my Atari 800 when I had little software and could not do too much with it. One of the first things that I did was buy a disk drive for the 800 as I found the tape to be just a little too flaky. That improved the situation a little, but at that point I only had 16K and by the time DOS was loaded, there was hardly any room left for anything else. So the next thing that I had to do was buy some more memory.

That whole package: the Atari 800, the extra memory, and the disk drive cost over $2000!! Then I went out and bought a colour TV set which has been dedicated to it since the day it was bought. The 1040ST with built-in double-sided, double-density disk drive and monochrome monitor are substantially less expensive than the original 800 package.

If this seems a little nostalgic, it is because, as you will have read, TAPS has decided to "wind down". For too long, too few have been doing too much of the work. The Executive, some of whom have been working for TAPS for over four years, have decided to call it quits.

Nor did I feel that I could continue to produce this newsletter almost by myself for ever. This may mean that I shall finally have time to write those programs that I have never had time to write, and when the current "panic" where I work is over, I may finally get around to exploring the ST.

Last month I wrote about "Freeware" as being a solution to software piracy. Unfortunately, as you will read in this issue, the Atari community was not mature enough for freeware. That is sad. As most of the owners of 6502-based Atari's know, there is almost nothing new being brought out for their machines at this time. What I have been telling you in my editorials for over two years is now a recognized industry fact. I invite everyone to read the article "June CES & the 8-bit Atari" starting on page 127 of the September ANALOG. This confirms on a wider scale what I have been telling you all along.

The summer issue of WireTAPS has become both the September issue and the last issue. In the almost two and a half years that I have been the Editor of this newsletter, I have learned a lot about word processing and I have been very pleased at the improvements in word processing software that have come out since I started. Now that this issue is done, I can look at the word processors which are available for the ST.

When someone at the Scarborough meeting in August heard that I had bought an ST, he wanted to know if I would be selling my 130XE and all of the related equipment. The answer is no. The 8-bit Atari line is the best deal on the market and at the moment I have too much invested in excellent software to want to give it up. In the long run, I may be using the ST to develop software for the 8-bit computers. I shall remain an 8-bit Atari user for a long time to come.

I really want to thank all of those TAPS' members who have taken time to contribute material for this newsletter, but again, time does not allow for everyone, I shall mention a few names however... Firstly, Sean Keely who has contributed so many excellent articles and reviews... Nick Coleman for his humourous reviews and articles, Douglas Skrecky for his support of Action! (sorry Doug I never was able to get Life to work.) John Hammond and Tim Glass for graphics covers. Graham Thorley and Brenda Tupling for news of their groups and Brenda for her crossword puzzles and articles. Please do not feel slighted if you name was not in this short list, it is just that I have so much to do at the moment and so little time. Thanks to everyone for your kind words and support.

DVC/65 C: AN UPDATE

By Peter BLAKE

In last month's issue, I reviewed the DVC/65 C compiler by Ralph E. Walden. Since then, I have received some good news and some bad news. First let me talk about the bad news.

I received a letter from Ralph Walden in which he writes:

> "...Unfortunately, shortly after I released DVC/65, someone typed in my entire documentation [62 8 1/2" by 11" pages. Ed.], removed all references to my name, and sent it out along with the other disks to various clubs and individuals claiming it was public domain. Accordingly, I am downplaying the freeware part, and in fact have placed copyright notices in CC, LINK, CEDIT, and FASTC. I also switched to a lighter colored paper, since dark paper was so ineffective a means of copy protection. You can add me to the list of programmers who will no longer develop commercial software for the 8-bit Atari because of piracy."

It is sad that the Atari community was not mature enough for Freeware. It is even sadder that that 8-bit Atari community has lost the services of Mr. Walden. It is a loss that the community can ill afford. I hope that someday, Mr. Walden may reconsider.

Now the good news. Mr. Walden is still supporting the DVC/65 C compiler and along with his letter, he sent me the latest release. Here are the changes:

1) As mentioned in his letter, the manual is now printed on lighter coloured paper. I think the shade is taupe, one of this season's "in" colours. This improves the legibility considerably.

2) DVC/65 now comes with separate run-time libraries so that you can run DVC/65 under any DOS that you wish. Indeed, I have tried it under SpartaDOS 3.2, DOSXL, and DOS 2.5 and it works extremely well. As this was my biggest reservation about DVC/65, I now have no reservations about it. If you are thinking of learning C, this is the compiler you should be thinking of getting.
3) Even if you wish to continue using DVC DOS, and it is still a good choice for developing C programs with, the run-time libraries will allow you to create versions of your programs to give to friends or to donate to your favourite public-domain library.
4) Finally, the editor program CEDIT has been enhanced to allow for "named" labels in addition to the use of line numbers with certain functions. This is an excellent improvement to an already superb editor for the C language. Indeed, I really do prefer using DVC/65's CEDIT to the editor which came with my $300 C compiler for the ST.

As for the lack of support for floating-point numbers, Mr. Walden writes: "... Putting floating point into the compiler would slow it down, since every variable would have to be checked for floating point along with all operators. But the final clincher is that I am very reluctant to spend several months developing floating point only to have it pirated and distributed on as wide a basis as DVC is. Not being independently wealthy, I need to generate an income from my programming efforts, or I won't be able to do them. So at this point I am concentrating my programming efforts on the ST and only small public-domain stuff for the 8 bit."

Finally I should mention that Mr. Walden pointed out that my benchmark SIEVE, which was based on the Action! version, still resembled Action! too much. By moving the declaration of FLAGS outside of the main () function, a significant gain in time would occur. I tried it, and indeed, the time with the optimized version was 1048 jiffies, as compared to 1175 jiffies for the way that I had originally written it.

small C -- AN OVERVIEW

By Peter BLAKE

This is the first and last of a series of articles which will introduce the programming language C. Actually, it will talk about a "subset" of C which is called "SMALL C". Small C does not have all of the power of full C but it is none-the-less a very powerful language. Unfortunately, to the best of my knowledge, there is no book on the market which deals exclusively with small C. I hope that this article will go a small way towards filling that gap. The reason that I am writing this is that a public-domain compiler, the ACE-C compiler is available. If you wish a copy of it, please contact me.

For those whose only programming language is BASIC, C may seem somewhat strange initially.

Firstly, whereas the RESERVED words in BASIC vary according to the version of BASIC that you use, the number of words in C is fixed, although not all versions of C have them all. This makes C at the same time more easy to learn and less easy to learn.

Secondly, the definition of the C language makes no provision for any input or output. To do that, a set of library function routines are used.

There are a few important things to remember about C before we begin. Atari BASIC is an "interpreted" language. That means that every time a BASIC statement is executed, a program called an interpreter is looking at your BASIC statement and figuring out what to do with the statement. C is a compiled language. You create a C source with a text editor. By convention, C source files usually have a file extender of .C. Once you have created a C source file, you then compile it and that turns your program into machine language instructions. Most C compilers have one more additional step called link-editing. What exactly is that? Because compiling is normally a very slow process, it is possible to create "libraries" of commonly used functions. If you do not compile these functions every time, you can save a lot of time. So these functions are kept in compiled form. But it now becomes necessary to Link-Edit or join the already compiled functions to the functions that you have just compiled. The output from the link editor is a stand-alone program which you can (load and) run. If this sounds confusing, we shall return to it later, and by then it will all be clear.

Rather than starting with the RESERVED words in C, it might make more sense to look at the structure or form of a C program.

```
Headers and definitions

Variable declarations

Function_1
Function_2
Function_3
    .
    .
    .
Function_n
```

That seems simple enough, doesn't it? What's the trick?? We'll see that later.

Headers are #include statements and definitions are #define statements. These are what are called **compiler directives.** A compiler directive has nothing to do with your program, but is your way of communicating with the program which is compiling your program.

The #include statement asks the compiler to copy some code from the default disk drive, this will usually be D1:, into the program. Most C programs begin with the cryptic statement:

```
#include <stdio.h>
```

The #define statement asks the compiler to substitute one string for another every time the string is used. Why? Because it is a more flexible way to write programs. Suppose for instance that you were writing a program with a table in it. The best way to do it is to define the size of the table in a define statement:

```
#define MAXSIZE 400
```

Then you test MAXSIZE to discover if the table is full, or whether you have reached the last possible entry you test MAXSIZE. If you then want to change the size of the table, you change the #define statement. All the rest of the code will then be self adjusting.

By the way, it is a C convention that defined words are capitalized. This helps you understand exactly what a program is doing. It is not a C requirement that these words be capitalized.

We now come to the part of the program called "Variable Declarations". Before we deal with them there is one thing that is very important to understand: There are two flavours of variables: global and local. Global variables are those which are defined here, that is outside of any function. A global variable is "known" to all functions in the program. Local variables are defined inside a function. It is possible in C to have a local variable and a global variable with the same name which can represent different values. In any case, unlike BASIC, **every variable in C must be declared before it can be used.**

So how, you are asking yourself, do you declare a variable? In small C there are two words which do it: int to define a two-byte integer and char to define a one-byte character. Both int and char are reserved words in C. Here are some sample definitions:

```
int number;
int number1, number2;
char letter;
```

from these examples we can gain an important fact of life about C: **every statement in C must end with a semi-colon.** Note that #define is a compiler directive and not a C statement. One letter is not terribly useful, but letters can be joined together in an array to become a string. Here is how a string is defined:

```
char word[10]; /* word is a string */
```

the above statement is the eqivalent of the Atari BASIC DIM WORD$(10). The "$" in BASIC is not necessary because the definition tells the compiler that "word" is to be treated as a string. We have also been introduced to a comment in C. A comment is any thing between /* and */. Some compilers allow for comments in the middle of a statement, other compilers consider anything after a /* to be a comment until the end of the line.

Now comes the fun: functions. Everything in C is a function. One function, called "main ()" receives control when the program is run. This means that every C program must have a main () function in it. A function looks like this:

```
function_name ( [parameters] )
[parameter declarations]
$(
  local variable definitions;

C statements or functions
$)
```

The "$(" and "$)" are the way that the C squiggles "{" and "}" are represented on the 8-bit Atari. "$(" is the C word for BEGIN and "$)" is the C word for END.

If the function has a parameter, the parameters are declared before the first "$(" in the function. In command-driven operating systems, like DOSXL and SpartaDOS, the main () function can be passed parameters. This allows you to write a program called COPY which you run by entering:

```
COPY FILEA p:
```

where the strings "FILEA" and "P:" would be passed as parameters to the main () function of your program COPY.

There are three main types of statements in C: assignment statements, program flow control statments and function invocation statements.

An assignment statement looks very much like an assignment statement in BASIC:

```
x = x + y / 100;
```

except that it has to end with the ";". In addition, C has some very useful "sort forms", which take a while to get used to.

```
++x;
```

is a C statement which is equivalent to BASIC's

```
x = x + 1;
```

and you can also use --x; for x = x - 1.

Although C shares many of the assignment operators with BASIC (= + - / *) there are more operators in C. "&", when used as a prefix, is equivalent to the BASIC ADDR() function. Thus the C statement:

```
x = &string;
```

is the equivalent of the BASIC statement:

```
1000 X = ADDR(STRING$)
```

Elements in an array are referenced with square brackets and not round brackets as in BASIC thus,

```
x = array[20];
1000 X = ARRAY(21)
```

are the same. Note that arrays in C usually start with the zeroeth element, whereas in BASIC, it is more usual to start with the first, although in Atari BASIC, there is a zeroeth element.

"%" yeilds the remainder (modulus) of the of the first operand by the second. Thus x = 7 % 5 gives a result of 2 in x.

C has a complete set of logical operators. && is used for a logical AND and || is used for a logical OR and ! is used for NOT. Remember that if you do not like && or ! because you cannot remember what they mean you can use #define statements such as:

```
#define AND &&
#define NOT !
#define ENDIF $)
```

```
#define WEND $)
```
at the beginning of all of your programs.

Be careful of ^ as it means exponentiation in BASIC and it is used for an exclusive OR in C.

The program flow statements are IF ELSE ELSEIF WHILE DO FOR. The C if statement is more powerful than the Atari BASIC IF because it allows for ELSE and ELSEIF. Here is what an if expression looks like:

```
if ( expression ) statement;
```

One of the joys (?) of C is that where ever you see "statement" you can always substitute $(statement; statement; ...statement; $).

The "expression" can be any valid C expression, normally however it is in the form of (var1 operator var2). The valid operators are usually:

```
==  (equals)
!=  (does not equal)
<   (is less than)
<=  (is less than or equal to)
>   (is greater than)
>=  (is greater than or equal to)
```

There are other operators which can be used, but if you are just beginning, it might be safest for a while to stick with these. The bibliography at the end will tell you about other sources where you can find out more about C. Here is a sample of an if statement in C.

```
if ( x == 1 ) x = 0;
elseif ( x == 0 ) $(
  x = 1;
  y = 0; $)
```

The next type of program control statement, which is not available in Atari BASIC is the WHILE statement. It looks like this:

```
while ( expression ) statement;
```

Here is a short example which might help you to understand better:

```
while ( ++i < 25 ) q = i * i;
```

As you might expect, 1 is added to i and if the result is less than 25, i is multiplied by i and the result is stored in q.

More commonly, the WHILE construct is used to control programs until the end of a file or of a string is reached.

```
while (( ch = getchar ()) != EOF ) $(
```

will read a file a character at a time and process it. Notice that the read "getchar()" is included as part of the expression. This economy of words is one reason why C is so popular. It is also one of the reasons that C programs are not easy for the casual reader to understand.

In the first while example, the value of i was what ever it was when the statement was first executed. As in BASIC, FOR allows the programmer to set the initial variable of the loop. The form of a FOR in C is:

```
for ( initialize; test; update ) statement;
```

Here is a simple example that shows how to write a for statement:

```
for ( x = 0; x <= 50; ++x ) $(
 printf ( "The square of %d is %d", x, x*x );
 printf ( "\n"); $) /* new line */
```

The final loop control is the DO ... WHILE, which is similar to WHILE except that the test is made at the bottom of the loop. Here is what the format of it is:

```
do $( statements
  $) while (expression);
```

C has some unusual words which change program flow: BREAK and CONTINUE.
BREAK causes a premature exit from a loop (FOR, DO or WHILE) and CONTINUE causes execution to go back to the start of the loop and continue missing all of the statements after the CONTINUE.

Well, I realize that this has been rather sketchy. Some examples will be worth a thousand words. First, here is an example which was taken from a book of C for the IBM PC. The program has been adapted for the 40-byte screen of the Atari.

```
/* to use enter a string */
/* to end hit <RETURN> */
/* */
main() $(
  int ch; /* read character */
  int index;
  int chnum;

  while ((ch = getchar()) != ' n') $(
    chnum = ch % 15; /* produces a number from 0 to
14 */
    index = 0;
    while (index++ < (18 - chnum)) $(
      putchar(' '); $)/* spaces to centre */
    index = 0;
    while (index++ < (2 * chnum + 1)) $(
      putchar(ch); $)/* print ch x times */
    putchar('\n');
  $) /* end of while */
$) /* end of main() */
```

Just to see how it works, enter KLMNOPQRSTULLL <RETURN>. Next, here is guessing game program which was originally written for the ST.

```
#define TRUE 1
#define FALSE 0

main() $(
  int num, guess, win, turns, play;

  play = TRUE;
  while (play) $(
    turns = 0;
    win = FALSE;
    num = get_num();
    while (!win) $(
      ++turns;
      guess = get_guess();
```

```
      win = check_guess(num,guess);
    $)
    printf("It took you %d turns.\n\n", turns);
    play = play_again();
  $)
$)

get_num() $(
  int n; /* local variable declaration */

  n = rnd(99);
  return(n+1);
$)

 get_guess() $(
  int g;

  g = 0;
  while (g<1 ||g>100) $(
    printf("Enter a number from 1 to 100: ");
    scanf("%d", &g);
    printf("\n\n");
  $)
  return(g);
$)

check_guess(num,guess)
int num, guess;
$(
  int wn;
  wn = FALSE;

  if (guess < num) $(
    printf("Too low n n"); $)
  else if (guess > num) $(
    printf("Too high n n"); $)
  else $(
    printf("You guessed it! n");
    wn = TRUE;
  $)
  printf("\n\n");
  return(wn);
$)

play_again() $(
  int ch, p;

  p = -1;
  /* ch = getchar(); */
  while ( (p!=TRUE) && (p!=FALSE) ) $(
    printf("Play again? ");
    if ( (ch=getchar()) == 'y' || ch == 'Y') $(
      p = TRUE; $)
    else if (ch == 'n' ||ch == 'N') $(
      p = FALSE; $)
  $)
  printf("\n\n");
  return(p);
$)
```

Well, space and time, those two enemies of this newletter have run out for the last time. I hope that you will get a copy of C and try to break out of the BASIC barrier. I wish that I could tell you that there was one good book to learn C with, but as yet, I have not found one. "C Primer Plus" by Waite, Prata and Martin, published by Howard W. Sams & Co. is not bad. The standard reference work on C is "The C Programming Language" by Kernighan and Ritchie published by Prentice Hall. I once saw a book on C that was under $20, and if you ever find one, I recommend that you buy it. As ever, keep computing.

TAPS TORONTO SEPTEMBER MEETING

The first meeting of TAPS Toronto will be held September 4th at Humbercrest School located at 14 ST Mark's Road.

To get there, travel north on Jane from the Jane Subway station to St.Mark's Road which is between Annette and Dundas streets. St Mark's runs west from Jane. There is parking at the back of the school for those who are driving.

A MESSAGE FROM THE TORONTO ATARI FEDERATION

It is with regret that we learn of the closing of the TAPS umbrella. Although the 'rivalry' between TAPS and TAF has, at times, been less than friendly, the existence of two groups does tend to keep both on their toes.

The philosophy of the groups has been different, TAPS going for smaller local groups and TAF for one group (now two as our ST group has its separate meeting).

We understand that the groups forming TAPS will still be operating as individual entities and wish them continued success. If we can help in any way, we invite the executive of these groups to contact us.

There is also an open invitation for members of the groups to come to any of the TAF meetings.

For information on dates and location, please phone our info-line, 223-9699, any time.

Louis P. Shimbart, Vice-President
for the TAF Executive

TAPS THEME LIBRARY DISKS

VOL	PROGRAM	DESCRIPTION	GROUP	SEC
TH01		***TAPS THEME LIBRARY - GAMES #1***		
TH01	TAPSINFO.DOC	NEW MEMBER INFORMATION MESSAGE	DOC.	008
TH01	ARUNMENU.SYS	AUTO-BOOT DISK MENU	SYSTEM	002
TH01	MENU	DISK DIRECTORY: SELECT & RUN PRGMS	UTILITY	068
TH01	ACROBAT	JUMP HURDLES & DUCK ARROWS	GAME	103
TH01	CHAOS.OBJ	SHOOT OBJECTS--HARDER THAN IT LOOKS	GAME	014
TH01	DSIO.OBJ	USED IN FILLUP	SUPPORT	001
TH01	FILLUP	FILL THE BOX WITH COLOUR--LIKE QIX	GAME	009
TH01	FILLUP.DAT	USED IN FILLUP	SUPPORT	026
TH01	FIREBUGS.OBJ	SHOOT BUGS BEFORE THEY HATCH	GAME	039
TH01	HOOKEY	Q*BERT TYPE GAME	GAME	051
TH01	HOPPER	FROGGER TYPE GAME	GAME	101
TH01	NITEFLYR	LAND AN AIRPLANE AT NIGHT	GAME	054
TH01	PLANET.OBJ	PLANETARY DEFENSE	GAME	027
TH01	ROBOT	BERSZERK TYPE ARCADE GAME	GAME	150
TH01	ROBOTSET.DAT	EXTRA DATA FOR ROBOT GAME	SUPPORT	009
TH02		***TAPS THEME LIBRARY - GAMES #2***		
TH02	TAPSINFO.DOC	NEW MEMBER INFORMATION MESSAGE	DOC.	008
TH02	ARUNMENU.SYS	AUTO-BOOT DISK MENU	SYSTEM	002
TH02	MENU	DISK DIRECTORY:SELECT & RUN PRGMS	UTILITY	068
TH02	BACTRION.OBJ	KILL BACTERION - SAVE BRAIN	GAME	040
TH02	BOPCON	CONSTRUCT EXTRA LEVELS FOR BOPOTRON	SUPPORT	036
TH02	BOPOTRON	MANY LEVELS AND LADDERS	GAME	078
TH02	GOBBLE	PACMAN TYPE EAT THE DOTS GAME	GAME	076
TH02	GRUBS	APPLE PANIC TYPE LADDER GAME	GAME	071
TH02	HATBATTL.OBJ	MOVE HEAD THRU MAZE TO GET HATS	GAME	042
TH02	HERBERT2	IMPROVED HERBERT-Q*BERT	GAME	087
TH02	LIVEWIRE.OBJ	FAST PACED ARCADE GAME	GAME	036
TH02	MONEY.OBJ	ARCADE QUALITY PICK UP THE MONEY	GAME	024
TH02	RACE.OBJ	ARCADE QUALITY SHOOT UP	GAME	033
TH02	RETROFIR.OBJ	LAND SPACESHIP ON ALIEN MOON	GAME	042
TH02	TWOGUN.OBJ	2-PLAYER FUN - SHOOT YOUR OPPONENT	GAME	019
TH03		***TAPS THEME LIBRARY - ADVENTURE #1***		
TH03	TAPSINFO.DOC	NEW MEMBER INFORMATION MESSAGE	DOC.	008
TH03	ARUNMENU.SYS	AUTO-BOOT DISK MENU	SYSTEM	002
TH03	MENU	DISK DIRECTORY: SELECT & RUN PRGMS	UTILITY	068
TH03	CASTLQST	GRAPHICS ADVENTURE	GAME	073
TH03	DUNDRAGN	DUNGEON & DRAGONS PLAYER CREATOR	UTILITY	097
TH03	SLAVE	CELLAR OF GOLGOLOTH ADVENTURE	GAME	170
TH03	STARTREK	CLASSIC STARTREK ADVENTURE GAME	GAME	184
TH03	WARGAMES	SIGN ONTO NORAD, AMAZE FRIENDS	GAME	065
TH04		***TAPS THEME LIBRARY - ADVENTURE #2***		
TH04	TAPSINFO.DOC	NEW MEMBER INFORMATION MESSAGE	DOC.	008
TH04	ARUNMENU.SYS	AUTO-BOOT DISK MENU	SYSTEM	002
TH04	MENU	DISK DIRECTORY: SELECT & RUN PRGMS	UTILITY	068
TH04	ADVENT5D	TEXT ADVENTURE - ANALOG 11	GAME	099
TH04	CRASHDIV.OBJ	ADVENTURE GAME FROM ANALOG #18	GAME	056
TH04	EROTIC	TEXT ADVENTURE WITH EROTIC OVERTONE	GAME	211
TH04	HEXAGON	GRAPHICS ADVENTURE	GAME	086
TH04	QUEST1	ADVENTURE GAME - NO DOCUMENTATION	GAME	130

VOL	PROGRAM	DESCRIPTION	GROUP	SEC
TH05		***TAPS THEME LIBRARY- COMMUNICATIONS***		
TH05	TAPSINFO.DOC	NEW MEMBER INFORMATION MESSAGE	DOC.	008
TH05	ARUNMENU.SYS	AUTO-BOOT DISK MENU	SYSTEM	002
TH05	MENU	DISK DIRECTORY: SELECT & RUN PRGMS	UTILITY	068
TH05	AMODEM44	TERMINAL SOFTWARE TO USE WITH MODEM	UTILITY	067
TH05	DIALER	PHONE LIST & TOUCH-TONE DIALER	UTILITY	019
TH05	DRIVER.DOC	DOCUMENTATION FOR DRIVER PROGRAMS	DOC.	176
TH05	HMDRIVER.OBJ	HOMEPAK DRIVER FOR MPP MODEM	UTILITY	023
TH05	MDRIVER.OBJ	MAIN DRIVER FOR MPP 1000 MODEM	UTILITY	018
TH05	SMDRIVER.OBJ	SMART MODEM DRIVER FOR MPP 1000	UTILITY	023
TH05	TSCOPE.DOC	DOCUMENTATION FOR TSCOPE	DOC.	047
TH05	TSCOPE.OBJ	TERMINAL PROGRAM FOR BIN. DOWNLOADS	UTILITY	087
TH05	TSCPLESS.DOC	DOCUMENTATION FOR TSCPLESS	DOC.	035
TH05	TSCPLESS.OBJ	MPP BINARY DOWNLOAD CONVERSION PRGM	UTILITY	078
TH06		***TAPS THEME LIBRARY - EDUCATIONAL***		
TH06	TAPSINFO.DOC	NEW MEMBER INFORMATION MESSAGE	DOC.	008
TH06	ARUNMENU.SYS	AUTO-BOOT DISK MENU	SYSTEM	002
TH06	MENU	DISK DIRECTORY: SELECT & RUN PRGMS	UTILITY	068
TH06	ANIMAL	ARTIFICIAL INTELLIGENCE DEMO	GAME	018
TH06	FUNCMACH	GUESS THE RULE FOR CHANGING NUMBERS	GAME	064
TH06	MATHTUTR	MATHEMATICS TUTORIAL	EDUC.	071
TH06	MCDONALD	CHILDREN'S EDUCATIONAL GAME	GAME	094
TH06	NATURE	SPRING NATURE QUIZ (ANALOG 11)	EDUC.	065
TH06	TENGAL	HANGMAN TYPE GAME - STAR WARS THEME	GAME	060
TH06	TYPING	BEGINNING TYPING LESSONS	EDUC.	049
TH06	VOCAB	TEST YOUR VOCABULARY	EDUC.	067
TH06	WORDSRCH	FIND THE HIDDEN WORDS	GAME	080
TH07		***TAPS THEME LIBRARY - UTILITIES***		
TH07	TAPSINFO.DOC	NEW MEMBER INFORMATION MESSAGE	DOC.	008
TH07	ARUNMENU.SYS	AUTO-BOOT DISK MENU	SYSTEM	002
TH07	MENU	DISK DIRECTORY: SELECT & RUN PRGMS	UTILITY	068
TH07	AUTOMATE	WRITE YOUR OWN AUTORUN.SYS FILES	UTILITY	018
TH07	ERRORS.LST	TEXT DESCRIPTION OF ERRORS	UTILITY	040
TH07	EXTBAS.DOC	DOCUMENTATION FOR EXTENDED BASIC	DOC.	094
TH07	EXTBAS.OBJ	PROVIDES 12 NEW COMMANDS TO BASIC	UTILITY	032
TH07	FIXXL.DOC	DOCUMENTATION FOR FIXXL	DOC.	012
TH07	FIXXL.OBJ	LETS XL/XE MACHINES RUN OLDER PRGMS	UTILITY	089
TH07	FNXNKEY	ADD FUNCTION KEYS TO YOUR KEYBOARD	UTILITY	015
TH07	FONT	GREAT CHARACTER SET EDITOR	UTILITY	078
TH07	LINELIST.LST	LIST PROGRAM LINE AT A TIME	UTILITY	004
TH07	LISTER	LIST PROGRAMS - EVEN CONTROL CODES	UTILITY	036
TH07	MFDUP	MULTIPLE FILE DUPLICATION	UTILITY	043
TH07	MULTCOLR	MULTI-COLOR SCREEN GENERATOR	UTILITY	030
TH07	MUSE.BAS	PROGRAMMING AID - BASIC VERSION	UTILITY	037
TH07	MUSE.OBJ	PROGRAMMING AID - ML VERSION	UTILITY	007
TH07	RENUMBER	RENUMBER LINES IN BASIC PROGRAM	UTILITY	013
TH07	STICKTST	GOOD JOYSTICK TEST ROUTINE	UTILITY	012
TH07	XLDOS	SPACE SAVER DOS ROUTINE FOR XL'S	UTILITY	018

TAPS THEME LIBRARY DISKS

VOL	PROGRAM	DESCRIPTION	GROUP	SEC
TH08		***TAPS THEME LIBRARY - LOGIC GAMES***		
TH08	TAPSINFO.DOC	NEW MEMBER INFORMATION MESSAGE	DOC.	008
TH08	ARUNMENU.SYS	AUTO-BOOT DISK MENU	SYSTEM	002
TH08	MENU	DISK DIRECTORY: SELECT & RUN PRGMS	UTILITY	068
TH08	ANIMAL	ARTIFICIAL INTELLIGENCE DEMO	GAME	018
TH08	BCONTRCT	BID FOR CONTRACTS AND BUILD BRIDGES	GAME	099
TH08	CONCEN	CONCENTRATION - MATCH THE SQUARES	GAME	045
TH08	MERLIN	LOGIC GAME - LIGHT UP ALL SQUARES	GAME	054
TH08	MSTRMIND	FIND THE COLOUR COMBINATION	GAME	027
TH08	POKER	DEAL THE BEST HANDS - ANTIC 03/84	GAME	085
TH08	QUATRAIN	DIFFICULT, MATCH THE PATTERNS,TIMED	GAME	057
TH08	REVERSI	(OTHELLO) PLAY COMPUTER OR OPPONENT	GAME	046
TH08	RICOCHET	LOGIC GAME - FIND HIDDEN OBJECTS	GAME	073
TH08	SCRAMBLE	RUBIK'S CUBE WITH LETTERS	GAME	050
TH08	TICTACTO	PLAY THE COMPUTER	GAME	036
TH09		***TAPS THEME LIBRARY - GRAPHICS***		
TH09	TAPSINFO.DOC	NEW MEMBER INFORMATION MESSAGE	DOC.	008
TH09	ARUNMENU.SYS	AUTO-BOOT DISK MENU	SYSTEM	002
TH09	MENU	DISK DIRECTORY: SELECT & RUN PRGMS	UTILITY	068
TH09	ATARI400	DRAW AN ATARI 400	DEMO	044
TH09	BALLSONG.OBJ	AMAZING GRAPHICS DEMO,BOUNCING BALL	DEMO	052
TH09	BOUNCE.OBJ	ACTION! BOUNCING DEMO - ANALOG 20	DEMO	030
TH09	CESDEMO	FANTASTIC DEMO FROM LAS VEGAS CES	DEMO	151
TH09	COLORS.256	DISPLAY ALL 256 COLOURS AT ONCE	DEMO	005
TH09	GR10DEMO	GR.10 COLOUR DEMO - RECTANGLES	DEMO	006
TH09	GRENHOLE	INTERESTING SPIRAL GRAPHICS	DEMO	011
TH09	GTIADEMO.GR9	?GR.9 COLOR DEMO- FAST & INTERESTIN	DEMO	004
TH09	KALDSCP.OBJ	EXTREMELY FAST ACTION! GRAPHIC DEMO	DEMO	059
TH09	LOGO	ATARI LOGO, WITH ROTATING COLOURS	DEMO	034
TH09	MOLECULE	3D MOLECULE LANDSCAPE	DEMO	009
TH09	OCT	GROUP OF GRAPHICS DEMONSTRATIONS	DEMO	064
TH09	RAINBOW	COLOUR GRAPHICS DEMO	DEMO	007
TH09	WTRFALL	DRAWS LANDSCAPE & WATERFALL	DEMO	029
TH09	XMASTREE.OBJ	A CHRISTMAS TREE THAT COSTS < $30	DEMO	083
TH10		***TAPS THEME LIBRARY - BUSINESS***		
TH10	TAPSINFO.DOC	NEW MEMBER INFORMATION MESSAGE	DOC.	008
TH10	ARUNMENU.SYS	AUTO-BOOT DISK MENU	SYSTEM	002
TH10	MENU	DISK DIRECTORY: SELECT & RUN PRGMS	UTILITY	068
TH10	CLOCK	CLOCK FACE WITH MOVING HANDS	DEMO	014
TH10	DATABASE	SMALL BUT VERSATILE DATABASE PRGRM	UTILITY	111
TH10	DATABASE.DOC	DOCUMENTATION FOR DATABASE	DOC.	040
TH10	DIALER	PHONE LIST & TOUCH TONE DIALER	UTILITY	019
TH10	INFBITS2	SIMPLE FLEXIBLE UBSTRUCTURED D'BASE	UTILITY	046
TH10	LJKDIR	READS LETTER PERFECT W.P. DIRECTORY	UTILITY	003
TH10	MORTGAGE	CANADIAN MORTGAGE CALCULATOR	UTILITY	056
TH10	MULTCOLR	MULTI-COLOUR SCREEN GENERATOR	UTILITY	030
TH10	SPSCRIPT.DOC	DOCUMENTATION FOR SPEEDSCRIPT	DOC.	136
TH10	SPSCRIPT.OBJ	BEST PUBLIC DOMAIN WORD PROCESSOR	UTILITY	067
TH10	TICKER.TAP	MOVING TICKER TAPE MESSAGE	UTILITY	011
TH10	TYPING	BEGINNING TYPING LESSONS	EDUC.	049

VOL	PROGRAM	DESCRIPTION	GROUP	SEC
TH11		***TAPS THEME LIBRARY - MUSIC #1***		
TH11	TAPSINFO.DOC	NEW MEMBER INFORMATION MESSAGE	DOC.	008
TH11	ARUNMENU.SYS	AUOT-BOOT DISK MENU	SYSTEM	002
TH11	MENU	DISK DIRECTORY: SELECT & RUN PRGMS	UTILITY	068
TH11	SOUNDGEN	CREATE & TEST DIFFERENT SOUNDS	UTILITY	036
TH11	BALLSONG.OBJ	AMAZING GRAPHICS & MUSIC DEMO	DEMO	052
TH11	EDITSOUN	SOUND EDITOR, CONTROL ALL 4 VOICES	UTILITY	064
TH11	FUGUE	FOUR PART FUGUE, WITH GRAPHICS	MUSIC	043
TH11	MERYXMAS	PLAYS 6 CHRISTMAS CAROLS	MUSIC	070
TH11	OCTOPUS	ED APOSTLE'S VERSION OF BEATLES HIT	MUSIC	095
TH11	PASSION.OBJ	INCREDIBLE MUSIC DEMO	MUSIC	093
TH11	PINAPPLE	SCOTT JOPLIN'S PINEAPPLE RAG	MUSIC	069
TH11	PUFF.MUS	DEMO OF POPULAR TUNE	MUSIC	031
TH11	SOUNDEMO	DEMONSTRATE VARIOUS SOUNDS	MUSIC	024
TH12		***TAPS THEME LIBRARY - GRAPHICS # 2***		
TH12	AUTORUN.SYS	AUTO-BOOT DISK MENU	UTILITY	037
TH12	SLIDESHO	DISPLAYS KOALA STYLE .PIC FILES	UTILITY	021
TH12	SLSHO130.OBJ	USE EXTRA MEM TO FLIP THRU PICS	UTILITY	023
TH12	TITLE.SYS	DISK TITLE FILE	SYSTEM	001
TH12	COMET.PIC	KOALA FORMAT SLIDESHOW FILE	PICTURE	021
TH12	SPACEPIC.PIC	" " " "	PICTURE	030
TH12	SHOE.PIC	" " " "	PICTURE	015
TH12	KAHN.PIC	" " " "	PICTURE	045
TH12	GHOSTPIC.PIC	" " " "	PICTURE	020
TH12	BRUCELEE.PIC	" " " "	PICTURE	037
TH12	DEADLINE.PIC	" " " "	PICTURE	026
TH12	CATHY.PIC	" " " "	PICTURE	035
TH12	JOANNE.PIC	" " " "	PICTURE	046
TH12	ZZTOP.PIC	" " " "	PICTURE	013
TH12	DISK.PIC	" " " "	PICTURE	008
TH12	EAGLE.PIC	" " " "	PICTURE	022
TH12	NASA.PIC	" " " "	PICTURE	019
TH12	STARWARS.PIC	" " " "	PICTURE	028
TH12	JACK.PIC	" " " "	PICTURE	024
TH12	APOLLO.PIC	" " " "	PICTURE	030
TH12	ROCKY.PIC	" " " "	PICTURE	050
TH12	DONKEY.PIC	" " " "	PICTURE	019
TH12	SPOCK.PIC	" " " "	PICTURE	013
TH12	CLOUSEAU.PIC	" " " "	PICTURE	062
TH12	RACER.PIC	" " " "	PICTURE	018

PROGRAM LIBRARY

VOL#	PROGRAM	DESCRIPTION	GROUP	SEC
D031		***TAPS SOFTWARE LIBRARY DISK D031***		
D031	MENU	DISK DIRECTORY: SELECT & RUN PRGMS	UTILITY	068
D031	PUZZLER	RECONSTRUCT PIECES OF PICTURE	GAME	079
D031	SAILBOAT.PIC	SAMPLE PICTURE FILE FOR PUZZLER	SUPPORT	062
D031	DOGRACE.OBJ	PLACE YOUR BETS ON GREYHOUNDS	GAME	169
D031	TITLES	GENERATES STAND-ALONE TITLE PROGRAM	UTILITY	111
D031	TITLES.DOC	DOCUMENTATION FOR TITLES	DOC.	012
D031	GHANDLER.OBJ	AUTORUN FILE TO INSTALL G: DEVICE	UTILITY	010
D031	AMSMUSIC.DOC	DOCUMENTATION FOR AMS MUSIC PLAYER	DOC.	011
D031	AMSXL.OBJ	AUTORUN FILE TO USE AMS ON XL/XE	UTILITY	012
D031	AMSPLAY	AMS MUSIC PLAYER	UTILITY	051
D031	GIRLFUN.AMS	SAMPLE AMS SONG	SUPPORT	072
D031	GHANDLER.DOC	DOCUMENTATION FOR G: DEVICE HANDLER	DOC.	007
D032		***TAPS SOFTWARE LIBRARY DISK D032***		
D032	AUTORUN.SYS	AUTO-BOOT DISK MENU PROGRAM	SYSTEM	012
D032	MENU.DOC	DOCUMENTATION FOR AUTORUN MENU PRGM	DOC.	015
D032	CREATE.OBJ	SUPPLY PARAMETERS TO CREATE MENU PR	UTILITY	019
D032	DIRSORT.OBJ	SORT DISK DIRECTORY ALPHABETICALLY	UTILITY	009
D032	ADDRESS	NAME & ADDRESS DATABASE DOES LABELS	APPLIC.	078
D032	ADDRESS.DOC	DOCUMENTATION FOR ADDRESS	DOC.	018
D032	PICLOAD	PICTURE LOADER - WORKS WITH G: DEV.	UTILITY	072
D032	PICLOAD.DOC	DOCUMENTATION FOR PICLOAD	DOC.	022
D032	SNOOPY.PIC	SAMPLE PICTURE FOR PICLOAD	SUPPORT	033
D032	SMALLABL	PRODUCE RETURN LABELS FOR MAILING	UTILITY	037
D032	VOYAGE.DOC	DOCUMENTATION FOR MILLION B.C. GAME	DOC.	072
D032	MILLION.OBJ	DAVID CASTELL'S ONE MILLION B.C.	GAME	164
D032	WORDADVN	CHASE LETTERS TO BUILD WORDS	GAME	074
D032	COPY130.COM	130XE WHOLE DISK COPY - ONE PASS	UTILITY	011
D032	DEMO130	SHOW USE OF BANK-SWITCHED 130XE MEM	DEMO	015
D033		***TAPS SOFTWARE LIBRARY DISK D033***		
D033	AUTORUN.SYS	AUTO-BOOT DISK MENU PROGRAM	UTILITY	012
D033	BANJOPIC.KER	PLAY CHORDS ON A BANJO	GAME	035
D033	DISKIO.DOC	DOCUMENTATION FOR DISKIO	DOC.	007
D033	THEMAID.BXL	FILE CLEANUP, MULTICOPY & DOS CMNDS	UTILITY	080
D033	DSKIOFIX.BAS	DOS 2.5 PATCH FOR ORIG DISKIO	UTILITY	013
D033	DSKIOMAK.BAS	CREATES NEW DISKIO PLUS AS AUTORUN	UTILITY	105
D033	MEDIEVAL.ERA	DAVID CASTELL'S TIME TRAVEL GAME	GAME	259
D033	SKYSCAPE	FIND STAR POSITIONS & HALLEYS COMET	APPLIC	112
D033	SPEEDSKI	SKI DOWNHILL SLALOM - AVOID TREES	GAME	030
D034		***TAPS SOFTWARE LIBRARY DISK D034***		
D034	AUTORUN.SYS	AUTO-BOOT DISK MENU PROGRAM	SYSTEM	012
D034	START	MICROCHECK TITLE SCREEN PROGRAM	SUPPORT	010
D034	MENU	MICROCHECK MAIN MENU PROGRAM	SUPPORT	019
D034	UTILITY	MICROCHECK UTILITIES PROGRAM	SUPPORT	064
D034	CHECKBOO	MICROCHECK CHEQUE ENTRY PROGRAM	SUPPORT	096
D034	CHECKPRT	MICROCHECK CHEQUE SEARCH PROGRAM	SUPPORT	067
D034	CHECKBAL	MICROCHECK CHEQUEBOOK BALANCING PGM	SUPPORT	057
D034	MICROAUT.OBJ	AUTORUN.SYS FILE FOR MICROCHECK DSK	SUPPORT	002
D034	PHOTO.OBJ	DIGITIZED PHOTO DEMO - GIRL WINKS	DEMO	076
D034	MICROCHB.DOC	DOCUMENTATION FOR MICROCHECK SYSTEM	DOC.	059
D034	BONK.OBJ	ARCADE STYLE GAME OF PACMAN GENRE	GAME	029
D034	DRIVER	CREATE ATARIWRITER PRINTER DRIVER	UTILITY	073
D034	BOINK.OBJ	ATARI 8-BIT POWER BOUNCING BALL	DEMO	078
D034	GUITARTU	USE COMPUTER TO TUNE YOUR GUITAR	APPLIC.	011

VOL#	PROGRAM	DESCRIPTION	GROUP	SEC
D035		***TAPS SOFTWARE LIBRARY DISK D035***		
D035	AUTORUN.SYS	AUTO-BOOT DISK MENU PROGRAM	SYSTEM	037
D035	MENU.DOC	DOCS RE CHRIS THOMSON'S MENU PRGM	DOC.	063
D035	TITLE.SYS	DISK TITLE FILE	SYSTEM	001
D035	SETPOKE.OBJ	DISPLAY RESULTS OF GRAPHICS CMNDS	UTILITY	063
D035	RAMDSK2D.COM	130XE RAMDISK FOR DOS 2.0 & DOS XL	UTILITY	007
D035	RAMDISK.DOC	DOCUMENTATION FOR RAMDISK 2D	DOC.	016
D035	DDIR.COM	DOUBLE COL DIRECTORY FOR SPARTADOS	UTILITY	002
D035	POWERHSE	BUILD FOOTBALL DYNASTY THRU SEASONS	GAME	136
D035	INCTAX85	1985 INCOME TAX AID - TI GENERAL	APPLIC.	147
D035	SUPERPNG.OBJ	VARIATIONS OF ORIG PONG TYPE GAMES	GAME	020
D035	HOCKEY.OBJ	AIR HOCKEY ARCADE TYPE GAME - FAST	GAME	074
D035	SPEEDCLC.OBJ	COMPUTE MAG. SPREADSHEET PROGRAM	APPLIC.	070
D035	SPEEDCLC.DAT	SAMPLE APPLICATION OF SPEEDCALC	SUPPORT	033
D036		***TAPS SOFTWARE LIBRARY DISK D036***		
D036	AUTORUN.SYS	AUTO-BOOT DISK MENU PROGRAM	SYSTEM	037
D036	TITLE.SYS	DISK TITLE FILE	SYSTEM	001
D036	LOADIT	LOAD ANYTHING WHETHER ML,LST OR SVD	UTILITY	054
D036	FORMATER.OBJ	VERY CONVENIENT FORMATTING UTILITY	UTILITY	008
D036	DEBUG.OBJ	M/L DEBUGGER & MEMORY DISPLAY	UTILITY	048
D036	DEBUG.DOC	LIST OF COMMANDS FOR DEBUG	DOC.	006
D036	NIGHTSHD	ANALOG TEXT ADVENTURE	GAME	101
D036	SOFTKEYS	AUTO NUMBER & 1-KEY CMDS IN BASIC	UTILITY	028
D036	CLASH.OBJ	ARCHON-STYLE M/L ARCADE GAME	GAME	043
D036	LANDERCS	LUNAR LANDER CONSTRUCTION SET	GAME	093
D036	SLIDESHO	DISPLAYS KOALA STYLE .PIC FILES	UTILITY	021
D036	SLSHO130.OBJ	USE EXTRA MEM TO FLIP THRU PICS	UTILITY	023
D036	PICPRINT	TAILOR B&W PIC DISPLAY BEFORE PRINT	UTILITY	058
D036	PICPRINT.DOC	DOCUMENTATION FOR PICPRINT	DOC.	021
D036	CALCDOC.DOC	COMMAND LIST FOR SPEEDCALC (D035)	DOC.	015
D036	TAXPRINT.LST	MERGE WITH INCTAX85 (D035) TO PRINT	UTILITY	005
D036	PAPRWGHT.OBJ	FIND SELF-DESTRUCT VECTOR APR. FOOL	UTILITY	012
D036	COYOTE.PIC	SLIDESHOW PICTURE FILE	SUPPORT	019
D036	GARFIELD.PIC	SLIDESHOW PICTURE FILE	SUPPORT	018
D036	ODIE.PIC	SLIDESHOW PICTURE FILE	SUPPORT	012
D036	YODA.PIC	SLIDESHOW PICTURE FILE	SUPPORT	021
D036	EAGLE.PIC	SLIDESHOW PICTURE FILE	SUPPORT	022
D037		***TAPS SOFTWARE LIBRARY DISK D037***		
D037	AUTORUN.SYS	AUTO-BOOT DISK MENU PROGRAM	SYSTEM	037
D037	TITLE.SYS	DISK TITLE FILE	SYSTEM	001
D037	GARDEN	PLAN YOUR VEGETABLE GARDEN	APPLIC.	103
D037	MUSICIAN	ANTIC PUBLIC DOMAIN MUSIC COMPOSER	APPLIC.	128
D037	MUSICIAN.DOC	DOCUMENTATION FOR MUSICIAN	DOC.	023
D037	SAMPLMUS	SAMPLE SONG FROM MUSICIAN	SUPPORT	020
D037	BASEBALL	2-JOYSTICK BASEBALL - HIT & FIELD	GAME	100
D037	VIDEOFIL	CATALOG YOUR VCR TAPE COLLECTION	APPLIC.	084
D037	VIDEOFIL.DOC	DOCUMENTATION FOR VIDEOFIL	DOC.	020
D037	TEXTCONV	PURGE TEXT FILES OF CAR.-RET. CHARS	UTILITY	007
D037	FILEINFO.OBJ	CHECK OBJECT FILE ADDRESSES, LENGTH	UTILITY	048
D037	TICTACTO	3-D TICTACTO - PLAYS FAST	GAME	096

PROGRAM LIBRARY

VOL#	PROGRAM	DESCRIPTION	GROUP	SEC
D038		*TAPS SOFTWARE LIBRARY DISK D038*		
D038	AUTORUN.SYS	AUTO-BOOT DISK MENU PROGRAM	SYSTEM	012
D038	CREATE.OBJ	CREATE YOUR OWN AUTO-BOOT MENU	UTILITY	021
D038	COL80.OBJ	80 COLUMN DRIVER - ATARIWRITER	UTILITY	013
D038	COL80.DOC	DOCUMENTATION FOR COL80	DOC.	017
D038	PASSWORD.LST	PASSWORD PROTECT YOUR DISKS	UTILITY	045
D038	PSWAUTO.OBJ	AUTORUN FILE FOR USE WITH PASSWORD	UTILITY	001
D038	BANNER	CREATE LARGE SIGNS - ANY PRINTER	UTILITY	032
D038	FONTDATA	SAVE CHAR. SET AS DATA STATEMENTS	UTILITY	019
D038	FONTLOAD	LOAD A CHARACTER SET IN BASIC	UTILITY	006
D038	FONTMAKE.OBJ	CHARACTER SETS - CREATE & MODIFY	UTILITY	021
D038	SPACE.FNT	SAMPLE FONT	SUPPORT	009
D038	STYLISH.FNT	SAMPLE FONT	SUPPORT	009
D038	BANK	PLANNER FOR BANKING ACTIVITIES	APPLIC.	054
D038	DATE	LIST OF DATES FOR BANK	SUPPORT	003
D038	WEEK	SUPPORT FILE FOR BANK	SUPPORT	004
D038	FIXED	FIXED EXPENSES FILE FOR BANK	SUPPORT	003
D038	BANK.DOC	DOCUMENTATION FOR BANK	DOC.	048
D038	DEBUT.OBJ	BASIC DEBUGGING UTILITY- 5 NEW CMDS	UTILITY	011
D038	DEBUT.DOC	DOCUMENTATION FOR DEBUT	DOC.	006
D038	ROCKSLID	LINE UP BOULDERS IN STRAIGHT LINE	GAME	063
D038	CLEAN.OBJ	COMMAND STACKER FOR DISK REORG	UTILITY	062
D038	CLEAN.DOC	DOCUMENTATION FOR CLEAN	DOC.	178
D038	PAGEFLIP	DEMO OF PAGE-FLIPPING IN BASIC	DEMO	014